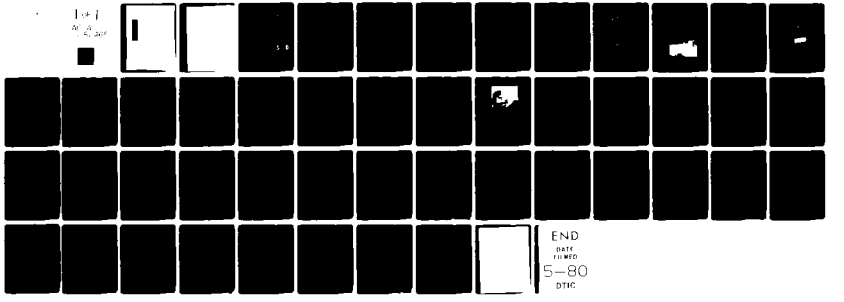
AD-A082 465 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
DIGIT: THE CURVE DIGITIZING SUBSYSTEM OF THE INTERACTIVE DATA D--ETC(U)
MAR 80 M B MARQUARDT

UNCLASSIFIED DTNSRDC-80/038 NL



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: DTNSRDC-80/038 (14)	2. GOVT ACCESSION NO.
	3. RECIPIENT'S CATALOG NUMBER (9)
4. TITLE (and Subtitle): DIGIT: THE CURVE DIGITIZING SUBSYSTEM OF THE INTERACTIVE DATA DISPLAY SYSTEM (6)	5. TYPE OF REPORT & PERIOD COVERED: Final Rept. Sep 77 — Apr 79
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Mary Beth Marquardt (10)	8. CONTRACT OR GRANT NUMBER(s): (12) 41
9. PERFORMING ORGANIZATION NAME AND ADDRESS: David W. Taylor Naval Ship Research and Development Center, Bethesda, Maryland 20084	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Program Element 62543N; Task Area SF43411301; Work Unit 1808-009
11. CONTROLLING OFFICE NAME AND ADDRESS: (16) F43411	12. REPORT DATE: Mar 80 (11)
	13. NUMBER OF PAGES: 2
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): (17) SF4341130I	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DTIC ELECTE S D MAR 3 1 1980 B

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Interactive Graphics
Digitizing
2-Dimensional

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

DIGIT is a high-performance interactive graphics program for use in extracting numerical data values (x,y-coordinates) from plots and drawings. It is a subsystem of the Interactive Data Display System (IDDS). In order to digitize, the user needs not only the program but also a digitizing tablet. IDDS, including DIGIT, is designed to run quickly and efficiently. Speed is obtained by minimizing the number of prompts. Each request performs a function. Efficiency is obtained by programming each function as a separate

(Continued on reverse side)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-LF-014-6601

UNCLASSIFIED 387683

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(Block 20 continued)

module, called a task. These tasks are loaded into core memory only as needed. Memory buffering techniques also save core storage.

The capabilities of DIGIT include digitizing, editing, plotting, and smoothing. Publication quality hardcopy is easily obtained from the Calcomp 936 or Tektronix 4662 plotters. IDDS and DIGIT are available for general use on the Tektronix 4014 graphics terminal connected to the CDC 6600 computer. The digitizing tablet is a Tektronix 4954 connected to the 4014.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

DIGIT is a high-performance interactive graphics program for use in extracting numerical data values (x,y-coordinates) from plots and drawings. It is a subsystem of the Interactive Data Display System (IDDS). In order to digitize the user needs not only the program but also a digitizing tablet. IDDS, including DIGIT, is designed to run quickly and efficiently. Speed is obtained by minimizing the number of prompts. Each request performs a function. Efficiency is obtained by programming each function as a separate module, called a task. These tasks are loaded into core memory only as needed. Memory buffering techniques also save core storage.

The capabilities of DIGIT include digitizing, editing, plotting, and smoothing. Publication quality hardcopy is easily obtained from the Calcomp 936 or Tektronix 4662 plotters. IDDS and DIGIT are available for general use on the Tektronix 4014 graphics terminal connected to the CDC 6600 computer. The digitizing tablet is a Tektronix 4954 connected to the 4014.

ADMINISTRATIVE INFORMATION

This work was performed under NAVSEA Mathematical Sciences Program, "Mathematical/Computational Technology for Advanced Ship Analysis and Design Techniques," Program Element 62543N, Task Area SF43411301, Task 14507, DTNSRDC Work Unit 1808-009. The NAVMAT cognizant program manager is Commander F. Hinton, MAT 08T23.

INTRODUCTION

A research mathematician wishes to compare curves on a graph in a report with the results from his latest theory. An engineer has a ten-foot long roll of test data showing certain vibration characteristics of a structure which he wants to analyze with a computer program. A naval architect has a large mechanical drawing of a ship hull showing twenty-five stations which are to be used to construct a computer model of the hull. All these investigators need a good digitizing system to produce meaningful numerical data from the different types of plotted curves available to them.

This report describes a Curve Digitizing Subsystem that is both easy to use and adaptable to many applications. It was developed to satisfy a need for a tool to digitize 2-dimensional curves and other data related to numerical fluid dynamics problems quickly and easily. However, it is general enough to be used in many other applications requiring digitized data (see Figure 1).

Curve Digitizing is a subsystem of the Interactive Data Display System (IDDS),* a collection of interactive graphics programs for displaying many types of data. A digitizing tablet connected to a graphics terminal is required in order to digitize. The programs are organized into subsystems for different applications, and include, in addition to the Digitizing Subsystem, special subsystems for contouring, for plotting seakeeping data, and for projecting 3-dimensional structures (e.g., ships, pressure vessels, aircraft, etc.). The core of the system is the UTILITY subsystem, which has tasks for system maintenance, array handling, 2-D plotting, and file manipulation, and which also provides access to the other subsystems. Some of the commands in the UTILITY subsystem are also available in the other subsystems, in particular, commands for array handling and 2-D plotting.

The entire Interactive Data Display System, including Digitizing, is designed on two levels, one for the novice user and one for the experienced user. The general concepts allow a user to accomplish a job with a minimum of instruction. A user with a greater knowledge of the computer and some of the workings of IDDS can take advantage of many additional capabilities which will be illustrated later in this report.

The system is based on modular programming which means that each command corresponds to a separate module or task. These tasks are self-contained, separately executable programs, which allow for easy modification, improvement, addition, and even replacement of an individual task. Thus IDDS is a dynamic system, constantly being modified, improved, and expanded.

---

*Mel Haas, Code 1843 is the developer of IDDS (to be documented).

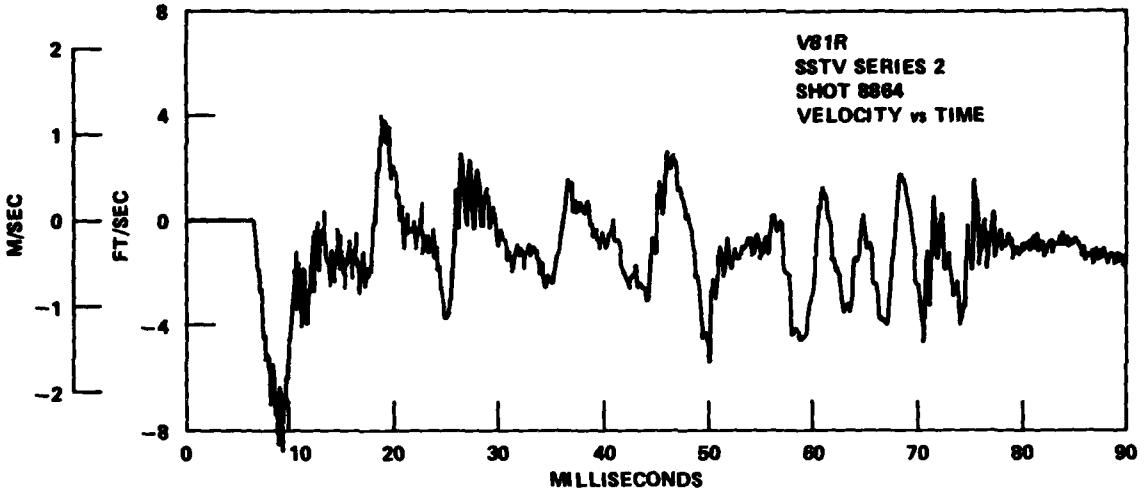


Figure 1a - Data to be Digitized

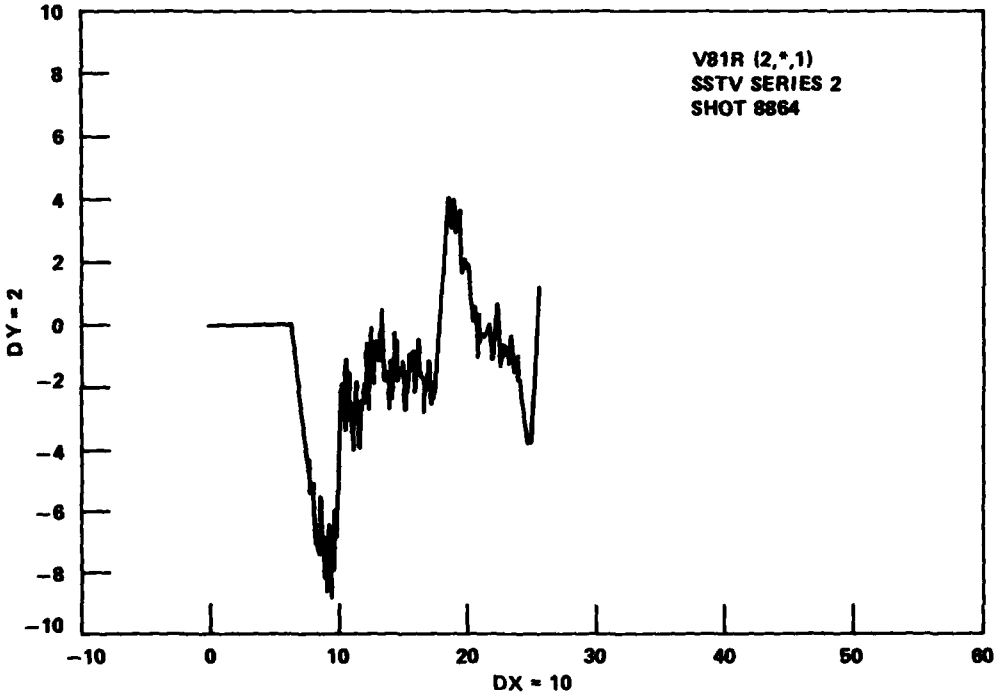


Figure 1b - Plotted Result

Figure 1 - Example of Data to be Digitized and Plotted Result

IDDS is a transaction-based system. It assumes the user knows what he wants to do and enables him to do it with very little prompting. Some information can be obtained on request, and error messages are provided. Minimizing prompting increases the efficient use of time and produces quicker response by the system because fewer interactions are required.

IDDS is available on the Tektronix 4014 Graphics Terminal connected to the CDC 6600 computer. The graphics tablet connected to the 4014 terminal is a Tektronix 4954. Figure 2 shows the equipment used with the Curve Digitizing Subsystem. Instant hardcopy is available at any time from the Tektronix 4631 unit which is also connected to the terminal. Higher quality hardcopy can be obtained by postprocessing to the Calcomp 936 or the Tektronix 4662 plotters.

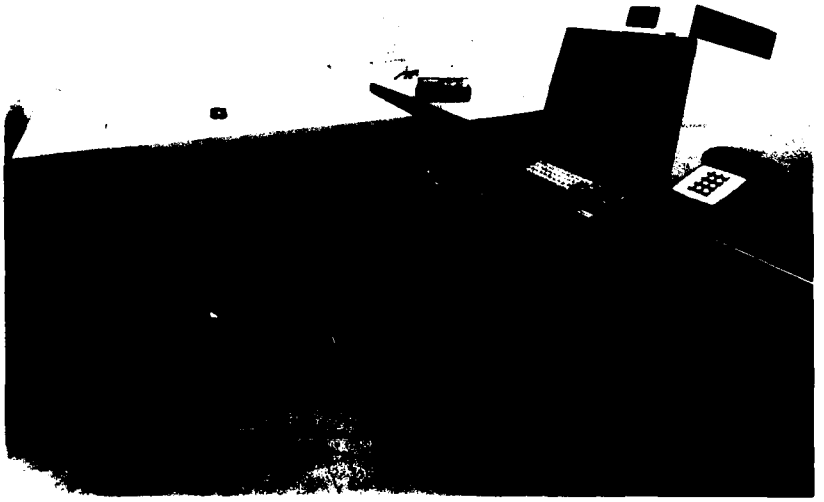


Figure 2 - Tektronix 4014 Terminal and Digitizing Tablet

Editing, plotting, and smoothing are the primary capabilities of the Curve Digitizing Subsystem, with additional capabilities of file handling, curve naming, and multiplication factor specification. Any number of curves, in any orientation, may be digitized. The only limitation is the physical size of the digitizing board. Arrays may be read in from a file and used as curve data, as long as the array dimensions are appropriate. All the capabilities of the Digitizing Subsystem are discussed in detail in the following sections of this report.

THE DIGITIZING TABLET

The graphics digitizing tablet used with the Curve Digitizing Subsystem is a Tektronix 4954, which is approximately 41 inches long by 34 inches wide (see Figure 2). The tablet inputs graphic data to the CDC 6600 computer through the Tektronix 4014 Display Terminal. The basic function of the graphics tablet is to convert the position of the cursor on the surface of a drawing or plot to a corresponding set of spatial (x,y) coordinates in digital form for use by the Display Terminal. The main components of the tablet are the digitizing surface, the cursor, and the power module.

Graphs, films, "menus", etc., may be placed on the surface of the tablet and then "digitized" as input data to the computer. Digitizing is used to mean both the digitizing of a point on a curve to yield x,y coordinates and the digitizing of a location on a menu to identify a command to be executed. Under the surface of the tablet are two grids of magneto-strictive wires, one set for the X-axis and one set for the Y-axis. An acoustic wave is sent along these wires and is detected by the cursor. The interval between the sending of the wave and its detection determines the identification of data points in the grid. The digitizing surface can be thought of as containing 1024 vertical lines and 1024 horizontal lines (wires). An identifiable location can be detected at the intersection of any vertical with any horizontal line. Identifiable points are 0.01 inch apart on the portion of the board which constitutes the valid data area. This ares is 38.4 inches in the X-direction by 30.72 inches in the Y-direction, providing a grid of 3840 by

3072 points. The Tektronix 4954 achieves this 0.01-inch resolution by operating with 12 bits of graphic data per point.

The Cursor, a small plastic device with crosshairs inscribed on the fact, contains a sensitive pick-up coil that detects the change in magnetic field caused by the acoustic wave. The detected signal is converted to digital information that directly relates to cursor position.

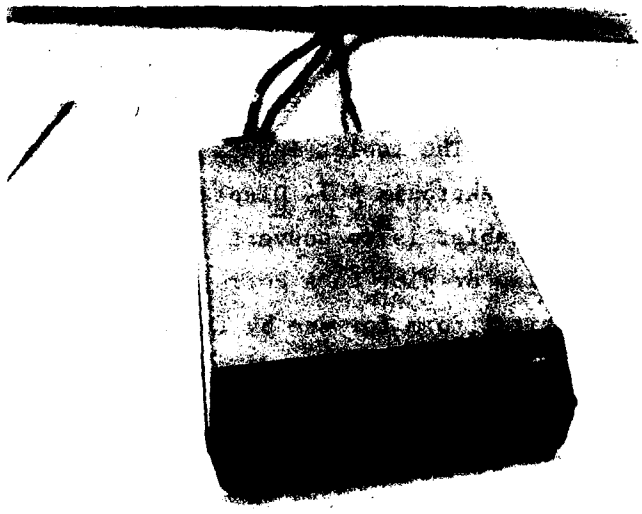


Figure 3 - Digitizing Tablet and Power Module

The Power Module, shown with the graphics tablet in Figure 3, contains the power supply and the connections which tie the graphics tablet components together. The front panel contains the power switch and indicators. The indicators provide the following information:

READY - "On" indicates the tablet is ready to send another point

PEN - Lights up when the cursor button is pushed

DATA - Lights up when data points are being transferred

POWER - Lights up when Power is on to both the terminal and the tablet.

OPERATING PROCEDURE

BACKGROUND INFORMATION

Although DIGIT provides only minimal automatic prompting, certain information about the system is available on request:

To display a list of all active commands in the subsystem as shown in Table 1, type the question mark symbol: ?

To obtain a brief description of any command and information about its parameters, type the command word followed by: ?

To obtain basic information about IDDS, type: HELP

In addition, the user must be familiar with the following procedures before starting to digitize:

To execute a control card statement within IDDS, type the symbol # before the control card statement. Note: The following control card statements can be executed directly (like a command word)

ATTACH
CATALOG
DISPOSE
FILES
UNLOAD

Control has been returned to IDDS when CONTROL CARD EXECUTED is printed on the screen.

To execute a predefined sequence of control card statements, called a CCL (Cyber Control Language) BEGIN/REVERT procedure, type the symbol: &. The available CCL procedures are described in Appendix A.

To execute a user-defined sequence of commands, called a "macro", as described in Appendix B, type the symbol: !

PRELIMINARY STEPS

Step 1. Tape the current menu of digitizing functions (Figure 4) and the curves to be digitized to the Tablet surface. Curves from more than one plot and/or more than one sheet of paper may be digitized during a work session. The only limitation on the number and size of the curves is the physical size of the Tablet surface.

TABLE 1 - LIST OF COMMAND WORDS

```
ACTIVE COMMANDS FOR LIST DGCMDS - CURVE DIGITIZING SYSTEM
ARBR       - READ ARRAYS FROM A BINARY FILE
ARBW       - WRITE ARRAYS ONTO A BINARY FILE
ARCR       - CREATE, MOVE OR RENAME AN ARRAY
ARDD       - DISPLAY DIRECTORY OF ARRAY DIRECTORIES
ARDIR      - DISPLAY OR CREATE AN ARRAY DIRECTORY
ARDL       - DELETE SPECIFIED ARRAYS
ARFR       - READ ARRAYS FROM A FORMATTED FILE
ARFW       - WRITE ARRAYS ONTO A FORMATTED FILE
ARND       - CHANGE OR CREATE ACTIVE ARRAY DIRECTORY
ATTACH     - ATTACH A FILE
CATALOG    - CATALOG A FILE
COMPRADC   - COMPUTE ROLL AMPLITUDE AND DECAY COEFFICIENT
CONTOUR    - ENTER CONTOURING SUBSYSTEM
COPY       - MAKE HARCOPY PRINT ON TEKTRONIX 4631
DGB1       - START A NEW ACTIVE CURVE
DGB2       - DISPLAY DIRECTORY OF CURVES
DGC1       - DEFINE USER COORDINATE SYSTEM
DGD1       - ADD POINTS TO CURVE SEQUENTIALLY
DGD2       - ADD AN EMPTY POINT
DGE1       - DELETE ONE POINT OR A RANGE OF POINTS
DGE2       - DELETE LAST POINT
DGE3       - INSERT ONE POINT OR A STRING OF POINTS
DGE4       - REPLACE A POINT WITH A POINT
DGF1       - DISPLAY X,Y POINT COORDINATES OF CURVE
DGF2       - PLOT THE CURVE
DGG1       - FIND USER COORDINATES FOR A DIGITIZED POINT
DGH1       - CHANGE TO SPECIFIED COMMAND SYSTEM
DGINITMNU  - INITIALIZE MENU
DGSMOOTH   - FIT CUBIC SPLINE TO DIGITIZED POINTS
DGXMULT    - CHANGE THE X ARRAY MULTIPLICATION CONSTANTS
DGYMULT    - CHANGE THE Y ARRAY MULTIPLICATION CONSTANTS
DIGITMENU  - SYSTEM COMMAND TO DIGITIZE A POINT ON THE MENU
DISPOSE    - DISPOSE A FILE
FILES      - DISPLAY LIST OF FILES LOCAL TO THE TERMINAL
HELP       - DISPLAY PAGES OF TEXT TO DESCRIBE ASPECTS OF THE IDDS SYSTEM
MEMO       - WRITE MEMO ON OUTPUT FILE
PAGE       - ERASE TERMINAL SCREEN
PLOT       - SAVE THE CURRENT PICTURE FOR POST PROCESSING
PQFIND     - DISPLAY PLOT QUICK COORDINATES OF CROSS
PQGRID     - CHANGE PLOT QUICK GRID INTERVALS
PQPLOT     - PLOT QUICK AN X ARRAY VS. A Y ARRAY
PQSCALE    - SCALE A PLOT QUICK PLOT
PQWINDOW   - CHANGE THE PLOT QUICK WINDOW ON THE DISPLAY
PQZOOM     - RESCALE PLOT QUICK PLOT WITH TRACKING CROSS
REDRAW     - REPEAT THE CURRENT PICTURE ON SCREEN
TDINIT     - CHANGE TO 3-D PERSPECTIVE DISPLAY SUB-SYSTEM
TIME       - DISPLAY DATE, TIME, CP TIME
UNLOAD     - UNLOAD SPECIFIED LOCAL FILES
UTILITY    - TRANSFER TO THE UTILITY SUB-SYSTEM
WEOF       - WRITE END-OF-FILE MARK ON SPECIFIED FILE
```

CURVE DIGITIZER MENU

A. Tape this menu on digitizing board and digitize the cross in the lower left corner, the upper left corner, and the upper right corner. If menu is moved, type DGINITMNU and redigitize crosses.

B. CURVE IDENTIFICATION :

{b1} Start a new active curve.
(For more information on b1 as a type-in command, type DGB1?)

{b2} Display directory of digitized curves.

C. CURVE AXES ORIENTATION :

{c1} Digitize three or more known points; then type in their values in the form: X1,Y1,X2,Y2,X3,Y3,.... These points should not lie on a line.

D. CURVE POINT INPUT :

{d1} Add points to the active curve in consistent order.

{d2} Add an empty point to the curve.

E. CURVE EDIT :

{e1} Delete one point or a range of digitized points (from 1st digitized point to 2nd digitized point or 1st typed index to 2nd typed index).

{e2} Delete last point in the curve array.

{e3} Insert one point or a string of digitized points or coordinate values (after a digitized point or typed index).

{e4} Replace a point with a point: digitize or type index of point to be replaced; then digitize or type coordinate values of replacement point.

F. CURVE DISPLAY :

{f1} Display x,y point coordinates of curve.

{f2} Plot the curve (scale with PQSCALE).

G. CURVE POINT VERIFICATION :

{g1} Find user coordinates of digitized point.

H. RETURN CONTROL :

{h1} Return control to the UTILITY system.

I. Curves may be saved with ARBW or ARFW.
(For more information, type ARBW? or ARFW?)

Figure 4 - Sample Menu

Step 2. Turn on equipment and log in (see Appendix C).

Step 3. Type the following to start execution of the Interactive Data Display System (IDDS):

BEGIN,MYPRO,,CAMV,IDDS.

The screen will be cleared and UTILITY SYSTEM will be displayed at the top of the screen.

Step 4. Type DIGIT to enter the Digitizing Subsystem.

Step 5. Initialize the menu (Figure 4), to establish the location of the menu on the digitizing surface so that the computer can recognize when a digitized point corresponds to a command on the menu or to a point on a curve. Neither commands on the menu nor curves may be digitized before the menu is initialized.

Place the crosshairs of the cursor over the crosses in the lower left corner, upper left corner, and upper right corner, in that order (as marked on the menu, 1, 2, 3), and press the button on the top of the cursor. The menu need not be reinitialized unless it is moved during the work session. If it is moved, type DGINITMNU. A message will appear on the screen telling the user to digitize the menu crosses again.

Step 6. To execute commands by digitizing, place the crosshairs of the cursor within the parentheses preceding the brief description of the command, and press the button on top of the cursor. Any command may also be executed by typing the letter and number prefixed with DG, e.g., DGB1. Some parameters available to a particular command, such as DGB1, may be input only by using that command as a type-in command. Digitizing and typing commands may be intermixed as suits the user.

COMMAND DESCRIPTIONS

GENERAL INFORMATION ON CURVES AND DIRECTORIES

Each curve array that has been digitized is located in a directory. Curves and directories are identified by short names (up to 10 characters) and long names (up to 100 characters). The user has the option of specifying these short and/or long names or using the default names, i.e., names that are automatically assigned *if user does not* provide one. The default short name for a curve is "CXXX", where XXX is the zero-filled,

right-justified, chronological number of the curve, e.g., C001 for the first curve defined on a directory. The default long name is the same as the short name. The default short name of the directory in the Digitizer Subsystem is DIGIT. The long name is DIRECTORY OF DIGITIZED CURVE ARRAYS. The default long name is the same as the short name.

The most recently defined curve is considered the active curve, and all functions are performed on the active curve.

The most recently defined directory is considered the active directory. A previously defined directory may be made active again by using the type-in command "DGB1". If no directory name is specified for a curve, the active directory is assumed. Any directory may be made the active directory by being specifically named with DGB1. For example, a directory created in another subsystem or by using ARRAY HANDLING Commands (see Appendix D) may be made active in the Digitizer Subsystem, but only by being given as the directory parameter in DGB1. This applies in particular when a curve read in from a file is to be the active curve to be recognized by the Digitizer Subsystem.

CURVE IDENTIFICATION

{b1} Start a new active curve.

This command provides more capability than the descriptive sentence implies. {b1} can define a new curve with a given or default name, reactivate a previously defined curve, rename a previously defined curve, make a curve read in from file or created at the terminal the active curve for digitizing, start a new active directory, or reactivate a previously defined directory.

When {b1} on the menu is digitized, the curve with the next default short and long name on the active directory (DIGIT if no other directory has been defined) will be defined. The first curve is automatically defined, as C001 on directory DIGIT. To specify a particular curve name and/or directory, DGB1 must be used as a type-in command.

DGB1 as a type-in command has the following format:

DGB1,CURVE NAME.DIRECTORY.DIRECTORY LONG NAME,
CURVE LONG NAME,OLD CURVE NAME.OLD DIRECTORY

When a curve short name is specified which does not exist on the active

directory, the curve is created and becomes the active curve. If the name does exist on the active directory, it becomes the active curve. When an existing curve name is given, it can be either a previously digitized curve or a curve array read in from a file. In either case, a long name may be specified. Any curve array from a file may become an active digitizing curve as long as it has the proper dimensions, e.g., (2,N,1), where N can be any number of points > 0.

To create a new directory, the user types DGB1,.DIR, which will define a curve C001 on directory DIR, and DIR will become the active directory. The curve short and long names may be specified, and the directory can optionally be given a long name.

The command "ARND,DIR" (see Appendix D) will make a specified directory DIR the active directory for other subsystems of IDDS, such as 2-D plotting or array handling, but DIR will not be the active directory for Digitizing. "DGB1,.DIR" must be specified to make directory DIR active for Digitizing. However, once a directory is the active Digitizing directory, it is also the active directory for IDDS.

An existing curve, either from a previous directory or from the current one, can be renamed with new short and/or long names. The newly named curve will become the active curve and the previous curve name will be deleted. A different directory can also be specified for the renamed curve, and this directory will then become the active directory.

No curves with zero (0) points are retained; that is, if a new curve is started before any points have been input to the previous curve, the previous curve name will be deleted. For example, if a user digitizes {b1}, then realizes he does not want the default curve name, he can use DGB1 as a type-in to give a specific curve name. The default curve name started previously will not be retained in the directory.

Whenever {b1} is used, either digitized from the menu or as a type-in command, a message is displayed on the screen at the end of the definition process giving the name of the active curve on the active directory. If an error occurs during the process, an error message will be displayed on the screen.

{b2} Display directory of digitized curves.

{b2}, as digitized on the menu, or DGB2, as a type-in command, will display on the screen a list of all the curves on the active directory. The format is:

Short name Number of points Long name.

The curves are listed in alphabetical, not chronological, order.

CURVE AXES ORIENTATION

{c1} Digitize three or more known points; then type in their values in the form: X1,Y1,X2, Y2,X3,Y3,... These points should not lie on a line.

The user must define the axis system of his curve(s) before any points on the curve can actually be digitized. When points on the curve are digitized, the coordinates are converted automatically to user coordinates. The user coordinate system need be redefined only if the next curve to be digitized is in a different axis system or if the curve in use is physically moved on the digitizing board.

The curve axes orientation is completely defined by three points which, when connected, do not lie on a straight line. As many as 16 points can be digitized to obtain more accuracy. It is generally best to spread the digitized points over the perimeter of the figure to be digitized to minimize the errors caused by skewed axes and wrinkled paper.

To define the curve axes, first digitize {c1} or type DGC1; second, digitize three or more points on the curve whose values are known; and, lastly, type in the actual coordinate values of the points digitized, in the same order in which they were digitized.

The digitization errors to be expected in the digitized X- and Y-values are displayed on the screen as information to the user. These errors are a function of actual hardware limitations and the range of the user coordinates and there is no way to eliminate or reduce them. If the range of user values lies between 0. and 10. in the X- and Y-directions, the maximum error in each will be of the order of .01. If the range of values is between 0. and .0001, the maximum error will be approximately 1.0E-7.

CURVE POINT INPUT

{d1} Add points to the active curve in consistent order.

Once a curve has been defined and the axis system established, the user is ready to digitize the points on the curve (see Figure 5). Mark the points on the plot to be digitized (on the paper) for editing purposes.

After digitizing {d1} or typing DGD1, digitize the individual points always proceeding in the same direction. Curves can be digitized from first point to last or last point to first, as long as the same direction is used consistently.

Part of a curve can be digitized, then edited, and then more of the curve can be digitized using {d1}. Added points are inserted at the end of the points array.

{d2} Add an empty point to the curve.

An empty point, indicating a gap in the curve, may be inserted at any time with {d2}. When {d2} is digitized or DGD2 is typed, a point with coordinates -1E30, -1E30 is set in the curve array. These coordinates are recognized in IDDS as a null point. The user can continue to digitize the curve with {d1}.

CURVE EDIT

Editing capabilities allow the user to delete points, insert points, and replace points. There may be several ways to accomplish an editing task, and the user should select the one that seems most straightforward.

{e1} Delete one point or a range of digitized points (from 1st digitized point to 2nd digitized point or 1st typed index to 2nd typed index).

{e1} provides two ways to delete one point or a range of points. A single point can be deleted by digitizing the point or by typing its index value. The index is the number assigned, in chronological order, to the points entered in the array. To delete a series of points, the user can (1) digitize the first point to be deleted, then the last point to be deleted (first and last referring to the order in which the points were digitized). All points between and including the digitized points

Figure 5 - Person Digitizing

will be deleted; or (2) type the index value of the first point to be deleted, then the index value of the last point. All points between and including the points with the typed indices will be deleted. Either method may be used after digitizing {e1} on the menu or typing DGE1. The index values may be determined from a chronological list of the curve points array which may be obtained with {f1}. An example of deleting points 5 thru 10 of the active curve, as a type-in, is *DGE1,5,10*.

{e2} Delete last point in the curve array.

{e2} will delete the last point in the curve points array, which may or may not be the last digitized point. This command is intended primarily for the correction of immediately recognized digitizing errors. There is no other input to {e2} - the user simply digitizes {e2} or types DGE2.

{e3} Insert one point or a string of digitized points or coordinate values (after a digitized point or typed index).

{e3} allows a user to insert one point or a string of digitized points or actual coordinate values. There are several ways this can be done. The user can either type the index value of the point after which values are to be inserted, or digitize that point on the curve. He can then either digitize the insertion points or type the actual coordinate values. For example, if the user wanted to insert the points (1.0,10.5), (2.0,15.3), (3.0,20.5), he could type DGE3,5,1.0,10.5,2.0,15.5,3.0,20.5, and those three points would be inserted into the array after the fifth point and the remainder of the points in the array would be moved down, i.e., their index values would be increased. Points may be inserted at the beginning of an array by typing DGE3,0,etc.

Typing and digitizing can be combined. For example, the user may type DGE3,1, then digitize the points to be inserted after the first point.

{e4} Replace a point with a point: digitize or type index of point to be replaced; then digitize or type coordinate values of replacement point.

{e4} allows the user to replace a point with a point. This can be done by typing, by digitizing, or by a combination as described for {e3}. Only one point can be replaced at a time, and with just one new point. If actual values are typed in, both the X- and Y-coordinates must be given, even if one coordinate value remains the same.

CURVE DISPLAY

{f1} Display x,y point coordinates of curve.

{f1} is used to display the point coordinates of the active curve, in chronological order. The format is:

Index number X value Y value.

Index numbers are adjusted according to the editing procedure used. Therefore, if a user wants to delete points and insert points (or vice versa) using the index numbers, he should be aware that the index numbers will be updated after the first procedure and perhaps he should use {f1} again before the second to be sure he is inserting or deleting after the

correct values.

{f2} Plot the curve (scale with PQSCALE).

{f2} is used to display a plot of the active curve. Scaling is determined automatically on the basis of the x,y coordinate values of the curve.

Scaling factors may be specified using the PQSCALE command (see Appendix E). The command is:

PQSCALE,XMIN,XDELTA,XMAX,YMIN,YDELTA,YMAX,
XINT,YINT,IRPT.

XMIN, XDELTA, XMAX correspond to the left value, the grid interval, and the right value of the horizontal axis. YMIN, YDELTA, YMAX correspond to the bottom value, the grid interval, and the top value of the vertical axis. XINT and YINT are the number of intervals on the grid. If negative values are used, grid lines will be drawn. Otherwise, tic marks along the axes mark the intervals. If all four values (MIN, DELTA, MAX, INT) are given for a particular axis, the axis is overdetermined and one or more values are ignored. Any parameters omitted will be calculated, based on the given parameters and/or the data. IRPT specifies whether to automatically replot (YES or null) or not (N,NO, or not YES).

Once scaling has been established, it remains the same until altered with a PQSCALE command. To return to atuomatic scaling, type PQSCALE with no parameters.

The PQPLOT command (see Appendix E) can also be used to plot, especially if the user wishes to superimpose several curves on one grid. To do this, the user types:

PQPLOT,XARRAY,YARRAY,N .

XARRAY corresponds to the first dimension of the short name of the curve, e.g., C001(1). YARRAY corresponds to the second dimension of the short name of the curve, e.g., C001(2). N says not to erase the previous plot.

CURVE POINT VERIFICATION

{g1} Find user coordinates of digitized point.

{g1} is available strictly for information. After the user digitizes {g1} or types DGG1, he can digitize a point in his coordinate

system and the user coordinates of that point will be displayed on the screen. No point is added to the curve array. Only one point may be digitized for information per command.

RETURN CONTROL

{hl} Return control to the UTILITY system.

{hl} returns control to the UTILITY Subsystem. The user must return to the UTILITY Subsystem before typing DONE to leave IDDS.

If {hl} is used as a type-in command, control can be transferred to any other subsystem in IDDS. The user types DGH1,--, with the two letters corresponding to the particular subsystem substituted:

CN for the Contouring Subsystem;
SK for the Seakeeping Subsystem;
UT for the Utility Subsystem; and
TD for the 3-D Subsystem*.

SAVING THE CURVES

In general, it is easier to digitize all curves before saving any. Save at one time only, at the end of the digitizing task. If it is necessary for the user to leave the terminal before he is finished digitizing all his curves, it is easier (and perhaps safer) to use &IDMFILE (see SAVING A WORK SESSION) to save his curve data than to catalog those curves already digitized.

Two ARRAY HANDLING Commands (see Appendix D) are available for writing the curve array information to file. One command is to write a binary file (ARBW) and the other is to write a formatted file (ARFW).

ARBW will write arrays to file in binary in a format to be read by the following FORTRAN code:

```
READ(...) NAME     # 1-10 characters
READ(...) LONG NAME     # 1-100 characters
READ(...) I,J,K,ARRAY     # dimensions and array
```

ARFW will write arrays to file in a format to be read by FORTRAN code:

```
READ(...,101) NAME     # 1-10 CHARACTERS
```

---

*Under development.

```
101 FORMAT(A10)
READ(...,102) LONG NAME    # 1-100 CHARACTERS
102 FORMAT(10A10)
READ(...,103) I,J,K    # ARRAY DIMENSIONS
103 FORMAT(3I4)
READ(...,104) ARRAY    # ARRAY(I,J,K) VALUES
104 FORMAT(6E13)
```

I will always be 2, J will be the number of pairs of coordinates, and K will always be 1. What this means to the user, basically, is that his curve points are stored as x,y pairs, J number of pairs, in an array dimensioned ARRAY(2,J,1). K will always be 1 since the curves are always two-dimensional.

ARBW and ARFW have the same parameters:

```
ARBW,LFN,A1,A2,A3,...,A10
ARFW,LFN,A1,A2,A3,...,A10
```

LFN is the local file name of the file onto which the arrays are written. LFN must be cataloged to save the file. If LFN is already a permanent file, it will be extended automatically.

Arrays are specified as: NAME.DIRECTORY. If no directory is specified, the current one is used. If no array is specified, all the arrays in the directory will be written. Up to ten different arrays and/or directories may be specified in one command. This does not mean that only ten arrays can be written to a file. Use as many Write commands as necessary to write all the arrays. The LFN may be the same for all.

It is very important to catalog the LFN. The user may do this with a CATALOG control card (also an available Command word) or by using a CCL (BEGIN/REVERT) procedure. The format is:

```
&SAVE,XXXX,LFN.
```

The user substitutes his four-letter user initials and the LFN used in the ARBW or ARFW command. The file will be cataloged as XXXX_LFN,ID=XXXX, with passwords PW=XXXX, XR=XXXX.

ADDITIONAL CAPABILITIES

DGSMOOTH

DGSMOOTH allows the user to smooth his curve. The command (not on the menu, type-in only) is:

DGSMOOTH,N,NAME.LONG NAME,RMSX,RMSY.

This command fits a cubic spline to the active curve with N interpolated intermediate points. The new smoothed curve will have the given short and long names or a default of the active curve name with a preceding "S". The smoothed curve is automatically displayed and the user may accept or reject the result. If it is accepted, the curve array will be added to the directory.

The smoothed curve will have a sum-of-squares deviation from the digitized points of NPTS*RMSX**2 in x and NPTS*RMSY**2 in y. If RMSX and/or RMSY are not given, the grid error (resolution) of the digitizer will be used. The resolutions in x and y are the values displayed on the screen at the completion of the definition of the user coordinate system, {c1}.

XMULT, YMULT

The two commands XMULT and YMULT allow the user to specify multiplication constants for the x and y values, respectively. The command parameters are the same:

XMULT,C0,C1,C2,C3

YMULT,C0,C1,C2,C3

where C0,C1,C2,C3 are substituted in the equation(s):

X = C0 + C1*X + C2*X**2 + C3*X**3

Y = C0 + C1*Y + C2*Y**2 + C3*Y**3

The default values are C0,C2,C3 = 0 and C1 = 1.

SAVING A WORK SESSION

If a user has not finished with a digitizing task but wishes to leave the terminal for a while, he can take a "snapshot" of his work at that moment by saving the IDMFILE. The IDMFILE is the Data Manager File which contains a record of work done up to that point. This allows the user to

come back later and resume work exactly where he left off. All previously defined curves will still be in previously defined directories and all previously defined digitizing parameters will still be in effect.

The IDMFILE is saved by cataloging with a CCL procedure. The user types:

&IDMFILE,XXXX,NAME ,

where XXXX is the four-letter user identification and NAME is some identifying name up to ten characters. The file is cataloged as:

XXXX_NAME_IDMFILE,ID=XXXX,PW=XXXX .

The lowest numbered cycle on the IDMFILE is automatically purged, if one exists.

When the user returns later to resume work, he begins IDDS by typing:

BEGIN,MYPRO,,CAMV,IDDS,ID,NAME ,

where ID is the ID under which the IDMFILE was cataloged and NAME is the identifying name specified. This will automatically begin executing IDDS, using the user's own IDMFILE.

The user will be in the Digitizing Subsystem if he was there before saving the IDMFILE. The menu need be reinitialized (by digitizing the three crosses) only if the menu was moved. The user coordinate system should be redefined, {c1}, if the paper containing the curves was moved or if a new coordinate system is to be used.

OBTAINING HARDCOPY

Instant hardcopy may be obtained at any time by pushing the hardcopy button on the terminal (above keyboard, to the right). If higher quality hardcopy is desired, use the PLOT command. This will save the characters to generate the plot on a permanent file device, which the user must catalog. If the file is cataloged after the first plot, any additional plots will extend the file automatically.

If the user just types PLOT, that plot is saved on a file with the default logical file name of PLOTF. PLOTF can be cataloged with one of two available CCL procedures:

&PLOT,XXXX or &SAVE,XXXX ,

where XXXX is the four-letter user identification. The file is cataloged

as XXXX PLOTF,ID=XXXX.

If the user wants to specify his own LFN, he types PLOT, LFN. This plot file may be cataloged with

&PLOT,XXXX,LFN

or

&SAVE,XXXX,LFN ,

where XXXX is the user ID and LFN is the same as for the PLOT command. The file will be cataloged as XXXX_LFN,ID=XXXX.

Either of these plot files may be postprocessed to produce hardcopy on the Tektronix 4662 or the Calcomp 936 (see Appendix F).

ACKNOWLEDGMENTS

The author would like to acknowledge Mr. Mel Haas for incorporating the smoothing capability into the Digitizing Subsystem, plus his numerous useful suggestions and encouragement. The author would also like to acknowledge Mr. Richard Van Eseltine for his ideas and efforts in the initial stages of developing and programming the Digitizing Subsystem.

APPENDIX A

CYBER CONTROL LANGUAGE (CCL) PROCEDURES

Cyber Control Language (CCL or BEGIN/REVERT) allows the user to manipulate control statements. Several CCL procedures are available to the user in IDDS. CCL procedures are executed within IDDS with &PROC, PARAMS. Outside of IDDS, they are available with:

BEGIN,MYPRO,,CAMV,PROC,PARAMS .

POSTPROCESS IDDS PLOTF FILE TO CALCOMP 936 TAPE:
&PP936,/ID/,/AC/,/CODE/,/BIN/,/MARGIN/,/SCALE/,/N1/,/N2/,/N3/,...,/N14/

ID - 4-character user identification
AC - Batch CHARGE card parameter (10 characters)
CODE - 4-digit user code
BIN - tape bin (2 letters and 4 digits)
MARGIN - left text offset (=0 none, =-1 no text, default = 25 characters)
SCALE - scale factor (=2 twice size, =1/3 one-third size, default = 1)
N1,N2,N3,...,N14 - plot selections (=+n plot n^{th} plot on file, =-n skip n^{th} plot on file, default = all)

This procedure will catalog the PLOTF file as ID_PLOTF, if it has not already been cataloged, then will submit a batch job to postprocess the file to the user's tape in BIN. The user must have a BIN tape located at the Central Site to use. The output will be printed at the Central Site, where the Calcomp Request card must be submitted at the end of the job.

CATALOG A FILE:
&SAVE,/ID/,/LFN/

ID - 4-letter user identification
LFN - logical file name

This procedure will catalog a file as ID_LFN for future use. If no LFN is specified, the PLOTF file is cataloged.

CATALOG PLOTF FILE:

&PLOTF,/ID/

This procedure will automatically catalog the PLOTF file as ID_PLOTF, ID=ID for future postprocessing.

SAVE THE IDDS DATA FILE:

&IDMFILE,/ID/,/NAME/

ID - 4-letter user identification

NAME - 1-10 character identifying name

This procedure will catalog the IDMFILE as ID_NAME_IDMFILE,ID=ID and will purge the lowest numbered cycle, if one exists.

APPENDIX B

MACRO CREATION

A macro is a series of commands to be executed automatically when !NAME is typed. The user indicates he wants to create a macro by typing !. IDDS prompts the user to type the NAME (and TITLE), the commands in sequence, and END to indicate termination of macro creation. Macros may have up to nine parameters, which are indicated in the commands !1,!2,...,!9, to indicate their chronological insertion from the command parameter sequence. For example, the user defines:

```
!A
PQP,!1,!2
PQP,!3,!4
END
```

To execute, the user types

```
!A,X1,Y1,X2,Y2
```

which will become

```
PQP,X1,Y1
PQP,X2,Y2 .
```

Concatenation can be used, e.g., CAEM!1PLOTF to insert the first input parameter, for specifying a particular permanent file name.

If the user wants to create a macro which is very long or complicated (subject to some error) or which he thinks he might want to modify at some time, he can use NETED to put his macro on a file which can be edited later. If the macro were typed in IDDS and some change was needed, the whole macro would have to be retyped.

Using standard NETED (see NETED documentation) procedure to set up the macro the way IDDS would expect to create it:

```
!
NAME TITLE
:
:
END
#IDDS.
```

Save the file and catalog.

To execute the macro creation, enter IDDS as usual (BEGIN,MYPRO,, CAMV,IDDS). Then ATTACH,LFN, your permanent file name with the macro commands. Next, type #IDDS,LFN, which will execute IDDS (# to execute a control card), from card images on a file, LFN. IDDS will execute again to create the macro. The #IDDS at the end of the macro will again execute IDDS interactively and the user can continue his work session, with his macro available. The macro can be saved within IDDS by saving the IDMFILE. This will eliminate the need to recreate the macro in subsequent sessions, unless a revised or new macro is needed.

NOTE: The NETED part of the procedure may also be done within IDDS, e.g., #NETED,LFN, etc.

APPENDIX C

GETTING STARTED

A standard procedure must be followed to start up the Tektronix 4014 terminal and enter IDDS:

A. Turn Power On:

1. Tektronix 4014 terminal
2. Hardcopy unit
3. Vadic modem (in back of modem box)
4. Digitizer power box (if using Digitizer Tablet)

B. Telephone 6600 Computer:

Dial X71004 for terminal in Room 118, Building 192.

When computer responds (a tone after 1 or 2 rings), lift white button on phone, but do not hang up receiver.

C. User LOGIN

Terminal will respond with standard Intercom introduction.

User types LOGIN,XXXXYYYYYY (where XXXX is user ID and YYYYYY is user Intercom letters).

The computer will then ask for account number.

When the account number is verified, computer will respond COMMAND-

D. Execute IDDS

User types BEGIN,MYPRO,,CAMV,IDDS to call the procedure to initialize IDDS.

When IDDS has begun execution, UTILITY SUBSYSTEM will be displayed on the screen.

The following specific command words provide entry to other subsystems of IDDS:

DIGIT	for Digitizing
CONTOUR	for Contouring
SEAKEEP	for Seakeeping.

Since details of this procedure may change somewhat from time to time (such as a change of extension for dialing the computer) the new user should check with Code 1843 if some problem arises.

APPENDIX D

ARRAY HANDLING COMMANDS

The following commands are available for reading, processing, and writing arrays:

ARBR	READ ARRAYS FROM A BINARY FILE
ARBW	WRITE ARRAYS ONTO A BINARY FILE
ARCR	CREATE, MOVE, OR RENAME AN ARRAY
ARDD	DISPLAY DIRECTORY OR ARRAY DIRECTORIES
ARDIR	DISPLAY OR CREATE AN ARRAY DIRECTORY
ARDL	DELETE SPECIFIED ARRAYS
ARFR	READ ARRAYS FROM A FORMATTED FILE
ARFW	WRITE ARRAYS ONTO A FORMATTED FILE
ARND	CHANGE OR CREATE ACTIVE ARRAY DIRECTORY

Reading Arrays

Reading and writing arrays on files are essential operations in the Interactive Data Display System (IDDS). The given list of commands has been designed to make these processes easy for the user.

There are two commands for reading arrays from files: ARBR, ARFR. The parameters are the same for both commands:

ARBR(ARFR), LFN, DIR.DIR LONG NAME, YES/NO.

This parameter sequence indicates that arrays are read from local file name LFN onto directory DIR (and optionally give the directory the specified long name), rewinding (YES) or not rewinding (NO) the file first. The file must have been attached previously, either before executing IDDS or in IDDS. The LFN of the Read command is the same as the LFN of the ATTACH command. If no directory name is specified, the arrays read will be located on a directory with the same name as the LFN. If no directory long name is specified, it will be the same as the short name. The file is automatically rewound first, unless NO rewind is specified (to read several records from the same file).

These Read commands expect to read formatted or binary arrays from a file which has been set up in a specific way. Designing the array WRITE's is the most important part of array handling (and usually the most difficult).

Setting Up Arrays On A File

ARBR (Binary Read) expects to read records from a file as written in FORTRAN:

```
WRITE(...) SHORT NAME      #1-10 character identifier of the
                           array(s)
WRITE(...) LONG NAME       #1-100 character identifier
WRITE(...) I,J,K,(((ARRAY(II,JJ,KK),II=1,I),JJ=1,J),KK=1,K)
```

The short name is the way an array is accessed, e.g.,

PQP, XARRAY, YARRAY

or

CNXYZ, XARRAY, YARRAY, ZARRAY .

The short and long names will generally appear on any plots produced.

The third WRITE statement has many variations, depending upon how the user wants his arrays set up.

(1) If they are X,Y pairs,

```
WRITE(...) 2,J,(X(JJ),Y(JJ),JJ=1,J)
```

which corresponds to X(1),Y(1),X(2),Y(2),X(3),Y(3),...

(2) If they are all X's, all Y's, all Z's

```
WRITE(...) I,3,(X(II),II=1,I),(Y(II),II=1,I),(Z(II),II=1,I)
```

or

```
WRITE(...) I,3,((ARRAY(II,JJ),II=1,I),JJ=1,3)
```

(3) If the user has a 2-D or 3-D array, not in X,Y pairs or X,Y,Z triplets, he could write them as two or three separate files, i.e., X array on one file, Y array on another file, and Z array on a third file. All three files could be placed on the same directory for ease of retrieval.

Example:

```
ARBR,X,MINE
ARBR,Y,MINE
ARBR,Z,MINE
```

(4) Several sets of array records may be put on one file. Each array may be 1-, 2-, or 3-dimensional.

Example:

```
XARRAY
XARRAY LONG NAME
5 2 1
```

<10 X Values>

YARRAY

Y ARRAY LONG NAME

5 2 1

<10 Y Values>

etc.

Or, such data could be consolidated into one array:

2DXYARRAY

2-D X AND Y ARRAYS

10 2 1

<10 X Values>

<10 Y Values>

(5) If an array is written simply as

WRITE(...) ARRAY

it will be read into a 1-dimensional array, e.g., ARRAY(N,1,1), where N is the total number of values read.

The user must understand how his arrays are stored in order to retrieve what he expects. It is best to set up the data in the most straightforward way, the way that is easiest to understand.

The easiest way to set up a file for ARFR (formatted read) is with NETED, with at least one space between values, in the format:

SHORT NAME

LONG NAME

I,J,K

<ARRAY VALUES>

The file may be written in FORTRAN with Formatted WRITE statements or in FREE FORMAT (see CDC FTN MANUAL).

The advantage of a formatted file is that it can be edited with NETED.

If ARFR, TERM is typed, arrays may be input at the terminal. A prompt is displayed to aid the user.

ARBR will try to read a file as if it were binary. If the file is formatted, a message will be displayed telling the user ARBR will try to read the formatted file. This is useful if the user is trying to use

data from a file whose format is unknown or uncertain.

Directories

All arrays read from file are automatically organized in directories. A short and/or long name may be specified for the directory in the Array Read Command. If no directory name is given, it will be the same as the LFN of the file. The long name will be the same as the short name. The last directory created is considered the active directory. When an array is specified as a parameter in a Command, it is assumed to be on the active directory if no other directory is given.

ARDD will display a directory of all the directories read in or created.

ARDIR, DIR.DIR LONG NAME will display a directory of all the arrays in the specified directory. If no directory name is given, the active (or current) one will be used. If a directory name is given, it becomes the active directory. If no directory exists with the given name, a new directory will be created.

If the user wishes to make a different directory the active one without listing the arrays, he can type ARND, DIR. The specified DIR will then become the active directory. If no DIR is specified, the name of the active directory will be displayed.

Creating and Deleting Arrays

ARCR, ANEW, NEW LONG NAME, AOLD will create a new array from an old one. This command can be used to change the dimensions of an array or create a copy of an array.

ARDL, A1, A2, A3,..., A10 allows a user to delete up to ten arrays. The arrays are specified as NAME.DIRECTORY. If no directory is specified, the active one will be used. If ARDL,.DIR is specified, all the arrays on that directory will be deleted.

Writing Arrays

Two commands, corresponding to the two commands to read a file, are used to write arrays to a file: ARBW, ARFW. ARBW writes a binary file and ARFW writes a formatted file. The parameters for the two WRITE commands are the same:

```
ARBW, LFN, A1, A2, A3,..., A10
ARFW, LFN, A1, A2, A3,..., A10
```

LFN is the file name onto which the arrays are written. As many as ten arrays may be specified at a time, in the form: NAME.DIRECTORY. More than ten arrays may be written to one file, but several Write Commands will be required. If no directory is specified, the active one is assumed. If only a directory name is given (A1 = .DIR), all the arrays in that directory will be written.

ARBW writes the arrays in a format to be read by FORTRAN codes:

```
READ(...) NAME
READ(...) LONG NAME
READ(...) I,J,K,ARRAY
```

LFN must be cataloged to be saved permanently. If LFN is already a permanent file, it will be extended automatically.

ARFW writes the arrays in a format to be read by FORTRAN code:

```
READ(...,101) NAME     # 10 characters
101 FORMAT(A10)
READ(...,102) LONG NAME     # 80 characters
102 FORMAT(8A10)
READ(...,103) I,J,K     # array dimensions
103 FORMAT(3I4)
READ(...,104) ARRAY     # ARRAY(I,J,K) values
104 FORMAT(6E13)
```

LFN must be cataloged as described for ARBW.

APPENDIX E

PLOT QUICK COMMANDS

PLOT QUICK enables the user to get 2-D plots quickly with a minimum of effort. The following commands are available to display 2-D plots and are displayed in alphabetical order:

PQARROW	ADD VELOCITY ARROWS TO A PLOT QUICK PLOT
PQFIND	DISPLAY PLOT QUICK COORDINATES OF CROSS
PQGRID	CHANGE PLOT QUICK GRID INTERVALS
PQPLOT	PLOT QUICK AN X ARRAY VS. A Y ARRAY
PQSCALE	SCALE A PLOT QUICK PLOT
PQWINDOW	CHANGE THE PLOT QUICK WINDOW ON THE DISPLAY
PQZOOM	RESCALE PLOT QUICK PLOT WITH TRACKING CROSS

The arrays may be set up on file or may be created at the terminal if the data are not extensive. Use ARFR or ARBR (depending on the format of the file) to read the data from file. ARDIR will list all the arrays on the file and the name of the directory. (See Appendix D, ARRAY HANDLING, for information on setting up and reading in files.)

The commands are discussed in the following paragraphs in the order in which they would logically be used.

PQPLOT

The basic plotting command is:

PQPLOT,XARRAY,YARRAY,IERASE,SYMBOL(INTERVAL),ISTYLE .

The XARRAY and YARRAY are identified by:

ARRAY SHORT NAME(I,J,K indices).DIRECTORY SHORT NAME .

The SHORT NAME is either the identifier specified on the data file or the default short name listed on the directory for the data file. The directory short name will be either the file name of the arrays or the directory name specified when the file was read. If no directory name is given, the active directory is assumed.

Examples of how to specify arrays in the PQPLOT command:

1. X,Y Pairs:

   If ARRAY(2,N,1)

   then PQP, ARRAY(1), ARRAY(2).

Note: PQP uniquely identifies the PQPLOT command, i.e., no other command begins with those three letters. All command words can be abbreviated in IDDS, as long as they uniquely identify the command.

2. All X, All Y:

If ARRAY(N,2,1)

then PQP,ARRAY(*,1),ARRAY(*,2).

Note: * indicates all the rows, columns, or planes.

3. Separate X and Y Arrays:

If XARRAY(N,1,1), YARRAY(N,1,1)

then PQP,XARRAY,YARRAY.

PQPLOT will plot a Y-array against its index values if no X-array is given, and will plot the first two dimensions of any array if no Y-array is given, e.g., example #1 could be just PQP,ARRAY. The X-array must always be specified for subsequent plots of different Y-arrays on the same plot.

A mesh or grid can be generated by plotting a 2-D X-array vs. a 2-D Y-array.

IERASE indicates whether to erase the screen before plotting (YES or null) or to superimpose (NO, N, or not YES).

SYMBOL identifies the plot symbol to be used for indicating the data points (any keyboard character except $,) ( ).

INTERVAL specifies the plotting symbol interval, within parentheses. If INTERVAL = 0, no plotting symbols will be used to denote the data points. If INTERVAL is a positive integer n, every n^{th} data point will be marked with a symbol. If INTERVAL = -n, data points will be marked starting with 1, and subsequently, every n+1. For example, if n=2, every 2nd data point will be marked with a symbol, i.e., 2, 4, 6, 8; if n=-2, data points 1, 3, 5, 7, etc., will be marked with a symbol.

ISTYLE indicates line style. A negative before the number indicates the points are not to be connected. 0 = solid line, 1 = dotted, 2 = dash-dot, 3 = dashed, 4 = long dashes, 9 = alternating short and long dashes.

PQPLOT with no parameters indicates a replot of the preceding plot, with perhaps new scaling. PQPLOT will scale to the data if no scaling has been specified. PQPLOT will also calculate grid intervals from the data if no gridding has been specified.

PQSCALE

PQSCALE,XMIN,DX,XMAX,YMIN,DY,YMAX,XINT,YINT,IRPT

is the command to scale. Scaling of the X- and Y-axes is given by three values: MIN (left or bottom value), DELTA (interval), and MAX (right or top value). Any two of the three values completely defines the scaling.

XINT and YINT indicate the number of intervals on the grid. If XINT or YINT is negative, grid lines will be drawn. The default is to draw tic marks along the axes to indicate the intervals.

IRPT = YES or null causes an automatic re-plot with new scaling. N or not YES inhibits replotting. PQSCALE with no parameters resets the default (automatic scaling). Scaling parameters will remain the same for all plots until changed or reset to defaults.

PQGRID

PQGRID,NIX,NIY

is a separate command to specify the number of grid intervals on the X- and Y-axes of the plot. A negative value will draw grid lines. This is the same as XINT and YINT in the PQSCALE command. PQGRID with no parameters resets the default of calculating grid intervals from the data, with no grid lines drawn.

PQWINDOW

PQWINDOW,LEFT,RIGHT,BOTTOM,TOP,AR,RX,RY

allows the user to change the physical display window on the screen. The default is a 14 x 10-inch frame represented by the absolute locations of 461,3940,100,2585 within the 0 to 4095 by 0 to 3120 display screen coordinates. The AR parameter allows the user to change the window by specifying an aspect ratio. The default is 1.4 based on the ratio 14/10 for the 14 x 10 frame. If a user wanted a 1/1 ratio, he would specify 1. for AR. RX and RY adjust the width and height, respectively, by ratios RX and RY. This command is concerned only with the physical size of the plot. The ratios have nothing to do with adjusting the actual user data. The default plot window (14 x 10) can be reset with PQWINDOW and no parameters.

PQZOOM

PQZOOM provides the option of rescaling the plot using the cross hairs to select a particular section of the plot to be enlarged to the whole plot size. Two methods can be employed. The first is to define a "box" around the area to be rescaled by indicating the lower left and upper right corners with the cross hairs (position the cross with the thumb wheels and type a space). The second method is to type PQZOOM,FACTOR where FACTOR is a numeric scale factor (e.g., 2.), and position cross at the center of the area to be the new plot.

PQFIND

PQFIND is strictly for information. The user positions the cross hairs on the point of interest and types a space (no carriage return required). The position of the cross in user coordinates will be displayed.

PQARROW

A more specialized and very useful capability is provided by PQARROW, which will add velocity arrows to a plot. The command is:

PQARROW,X,Y,U,V,RATIO,LOC,HL,HW,ISTYLE .

X,Y are the arrays of position.

U,V are the arrays of velocity in the X,Y direction.

Arrays are specified in the form:

NAME(I,J,K).DIR (as explained under PQPLOT).

If Y, U, V are not given, the first four elements of the smallest dimension of the X-array will be used for X, Y, U, V. If Y is not given, the first two elements of the smallest dimension of X will be used. The same is true for arrays U and V.

RATIO is an optional parameter for specifying the length of a unit velocity (in X,Y units). LOC gives the location of the velocity arrow relative to the actual data point, e.g., default = 0, X,Y at tail, = 1., X,Y at head, = .5, X,Y at midpoint. HL and HW are the arrowhead length and width (fractions of arrow length). ISTYLE indicates the line style for the arrow, i.e., default = 0, solid line, = 1, dotted, = 2, dash-dot, = 3, dashed, = 4, long dashes, etc.

PQPLOT must be called first to establish a grid and axis markings. PQPLOT,X,Y,,,-1 may be used to display an empty frame.

APPENDIX F

HARDCOPY DEVICES

CALCOMP 936

The CCL procedure for postprocessing the PLOTF file to Calcomp tape assumes the file is cataloged as ID_PLOTF,ID=ID. To begin the procedure, type (at a terminal):

```
BEGIN, MYPRO, , CAMV, PP936, ID, AC, CODE, BIN, MARGIN, SCALE,
                N1,N2,...,N14.
```

ID = 4-character user identification

AC = Batch charge card parameter (10 characters)

CODE = 4-digit user organization code

BIN = tape bin identification (2 letters and 4 digits)

MARGIN = left text offset from 4014 screen

= 0, text will be printed

= 1, no text

default = 25, text offset 25 character widths to left

SCALE = scaling factor (=2, twice size; 1/3, one-third size; default =1, same size as terminal)

N1,N2,...,N14 = plot selections (=+, plot; =-, skip; default = all)

This procedure submits a batch job to process the information on the file onto a tape for Calcomp. The output will be printed at the Central Site and a Calcomp request card must be submitted there to get the final hardcopy. Figure 6 shows a sample Calcomp Request Card.

TEKTRONIX 4662

The 4662 hardcopy unit is located in Building 193 (Burroughs Building, DTNSRDC). It is accessed through INTERCOM. The advantages of using the 4662 include good quality, choice of pen color (black, blue, red), and quick completion of the finished picture.

The user must make sure all needed plugs are connected, turn the unit itself on, and dial into the computer (6600) with the standard INTERCOM sign-on. The user types in the CCL procedure that drives the 4662:

```
BEGIN, MYPRO, , CAMV, PP4662, ID, NAME
```

where ID is the 4-letter user identification and NAME is the name used in the &PLOTF or &SAVE procedure. A prompting comment will be displayed at

CMLD CALCOMP / COM REQUEST SECRET CONFIDENTIAL PRIVATE OFFICAL USE

DATE	JOB ORDER #	USER NAME	USER ID	CODE	EXT

SYSTEM CDC B7700	TAPE ID	DENSITY 200 556 800 1600	TRACKS 7 9	LABELLED YES NO	DATE RUN

CALCOMP			COM	TOTAL TIME
STD SETUP YES NO	PAPER #	ORIGIN X____ Y____	COM PROCEDURE NAME	TOTAL FIC /FRMS
START BLOCK	STOP BLOCK	NO BLOCKS/FILES	FORM FLASH STD NONE _____	OPERATOR/ COMMENTS
PEN 1 2 3 COLOR		EST TIME	NO OF FILES NO OF FICHE _____ _____	

SPECIAL INSTRUCTIONS

Figure 6 - Calcomp Request Card

the user terminal

PREPARE PLOTTER PAPER, PEN, MARGINS, ETC.

TYPE PP4662, , MARGIN, SCALE, N1, N2, ...

There is usually "magic paper" available at the plotter with the right and left margins marked to produce a 1-to-1 correspondence with the Tektronix 4014 screen. Two "dots" on the paper indicate the UPPER RIGHT and LOWER LEFT points to be set. Remove the pen cap and lower the pen (optionally change to a different color). Locate the LOWER LEFT with the "joy" stick (bottom right side of plotter). When the pen is over the dot, SET the LOWER LEFT. Then use the stick to move the pen to the UPPER RIGHT and SET it likewise. Lift the pen. Remove the "magic" paper and insert a clean sheet, being careful to smooth it. Lower the pen and type

PP4662,, MARGIN, SCALE, N1,N2, ...

as described for the Calcomp 936. The terminal will prompt the user to CONTINUE or STOP after each plot is completed. A new sheet of paper must be inserted for each new plot. After completing all the plots, the user logs out and turns off the equipment.

INITIAL DISTRIBUTION

Copies	
3	CHONR
	1 ONR 430/R.J. Lundegard
	1 ONR 430B/M.Cooper
	1 ONR 438/R.D. Cooper
3	NRL
	1 J. Boris
	1 J. Hansen
	1 Library
1	USNA/Tech Lib
1	NAVPGSCOL/Lib
1	NAVWARCOL
1	ROTC, MIT
1	NUSC/Newport
	1 3635/Carol Hervey
5	NAVSEA
	1 SEA 032
	1 SEA 321
	1 SEA 32R/T.E. Peirce
	2 SEA 966/Lib
1	NAVSHIPYD BREM/Lib
1	NAVSHIPYD CHASN/Lib
1	NAVSHIPYD MARE/Lib
1	NAVSHIPYD NORVA/Lib
1	NAVSHIPYD PEARL/Lib
1	NAVSHIPYD PHILA/Lib
1	NAVSHIPYD PTSMH/Lib
12	DTIC

Copies	
2	ORI, Inc. 1400 Spring Street Silver Spring, MD 20910 1 Dana Gentile 1 Claude Williams

CENTER DISTRIBUTION

Copies	Code	Name
1	15	W.B. Morgan
2	1544	N. Stark D. Nigro
4	1552	J.H. McCarthy, Jr. N. Groves F. Thrasher J. Libby
1	1556	J. Alvarado
9	1568	S.L. Bales T. Applebee E. Foley L. Motter J. Voelker H. Jones W. McCreight W. Meyers R.S. Peterson
1	16	H.R. Chaplin
1	1619	P.S. Montana
1	17	W.W. Murray
1	172.3	P.N. Roth
1	18	G.H. Gleissner
1	1802.1	H.J. Lugt
1	1802.2	F.N. Frenkiel
2	1809.3	D. Harris
1	1820	A.W. Camara
1	1840	J.W. Schot
1	1843	H.J. Haussling

Copies	Code	Name
50	1843	M.B. Marquardt
11	1843	M. Haas R.T. VanEseltine J. Telste R. Coleman M. Marcus S. Ohring J. Dean R.L. Shoaff L. Brunson C.B. Franks L. Sander
1	1844	S.K. Dhir
5	1844	G. Everstine F. Henderson M. Hurwitz M. Golden A. Quezon
1	1850	T. Corin
1	1870	M. Zubkoff
1	1890	G. Gray
1	19	M.M. Sevik
10	5211.1	Reports Distribution
1	522.1	Unclassified Lib (C)
1	522.2	Unclassified Lib (A)